

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - MySQLWeek5/src/foods/Appfoods.java - Eclipse IDE
- File Menu:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Package Explorer:** Shows the project structure:
 - final project
 - MyFirstProjects
 - MyProject
 - MyProject1
 - mysql
 - MySQLWeek5
 - JRE System Library [JavaSE-1.8]
 - src
 - foods
 - Appfoods.java
 - foods.dao
 - DBConnection.java
 - Food.java
 - foodDao.java
 - foods.types
 - Food.java
 - foods.mysql
- Code Editor:** Displays the Java code for `Appfoods.java`. The code handles user input for food operations like creation, alteration, and deletion.
- Outline View:** Shows the class hierarchy and methods for `Appfoods`, including `scanner`, `FoodDao`, `Food`, and various `run` and `delete` methods.
- Bottom Status Bar:** Shows the current file is `Appfoods [Java Application]`, the Java runtime is `C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe`, the date is `(Nov 1, 2020 2:36:01 PM)`, and the connection status is `Connection successful !!!!!!!`.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "MySQLWeek5".
- Appfoods.java:** The active file contains Java code for a food management application. It includes methods for listing foods, deleting foods, modifying foods, and altering foods.
- Task List:** Shows a list of tasks related to the current file.
- Outline:** Displays the class hierarchy and member list for Appfoods.
- Bottom Bar:** Includes tabs for Problems, Javadoc, Declaration, Console, and Coverage. The Console tab is selected, showing the command "javaw.exe" and the date "Nov 1, 2020 2:36:01 PM".

```
final project
MyFirstProjects
MyProject
MyProject1
mysql
MySQLWeek5
JRE System Library [JavaSE-1.8]
src
  foods
    Appfoods.java
  foods.dao
    DBConnection.java
    FoodDAO.java
  foods.types
    Food.java
    foods.mysql
Referenced Libraries
  practice
    practices
    school
    week4
    week4Coding
    week5
    week6
    weeksSQL
```

```
final project
MyFirstProjects
MyProject
MyProject1
mysql
MySQLWeek5
JRE System Library [JavaSE-1.8]
src
  foods
    Appfoods.java
  foods.dao
    DBConnection.java
    FoodDAO.java
  foods.types
    Food.java
    foods.mysql
Referenced Libraries
  practice
    practices
    school
    week4
    week4Coding
    week5
    week6
    weeksSQL

Appfoods.java
package com.appfoods;
import java.util.Scanner;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

public class Appfoods {
    private static Connection connection;
    private static PreparedStatement ps;
    private static ResultSet rs;
    private static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        try {
            connect();
            listFoods();
        } catch (SQLException e) {
            System.out.println("Warning Error: " + e.getMessage());
        }
    }

    private static void connect() throws SQLException {
        String url = "jdbc:mysql://localhost:3306/appfoods";
        String user = "root";
        String password = "root";
        connection = DriverManager.getConnection(url, user, password);
        System.out.println("Connection successful");
    }

    private static void listFoods() {
        System.out.println("List of Foods");
        System.out.println("1. List Foods");
        System.out.println("2. Add New Food");
        System.out.println("3. Delete Food");
        System.out.println("4. Modify Food");
        System.out.println("5. Alter Food");
        System.out.println("6. Exit");
        int choice = scanner.nextInt();
        switch (choice) {
            case 1:
                listFoods();
                break;
            case 2:
                addFood();
                break;
            case 3:
                deleteFood();
                break;
            case 4:
                modifyFood();
                break;
            case 5:
                alterFood();
                break;
            case 6:
                System.out.println("Thank you for using our food management system!");
                System.exit(0);
            default:
                System.out.println("Invalid, try again");
        }
    }

    private static void addFood() {
        System.out.println("Enter the Food Name");
        String name = scanner.nextLine();
        System.out.println("Enter the Food Description");
        String description = scanner.nextLine();
        System.out.println("Enter the Food Price");
        double price = scanner.nextDouble();
        FoodDAO.addFood(name, description, price);
        listFoods();
    }

    private static void deleteFood() throws SQLException {
        listFoods();
        System.out.println("Enter the Food Id that you want to delete:");
        int id = (int) Long.parseLong(scanner.nextLine());
        System.out.print("So sad!");
        FoodDAO.deleteFood(id);
        listFoods();
    }

    private static void modifyFood() throws SQLException {
        listFoods();
        System.out.println("Enter the Food Id that you want to modify:");
        int id = (int) Long.parseLong(scanner.nextLine());
        System.out.print("Enter the new Food name");
        String name = scanner.nextLine();
        FoodDAO.modifyFood(id, name);
        listFoods();
    }

    private static void alterFood() throws SQLException {
        listFoods();
        System.out.println("Enter the Food Id that you want to modify:");
        int id = (int) Long.parseLong(scanner.nextLine());
        System.out.print("Enter the new Food name");
        String name = scanner.nextLine();
        FoodDAO.alterFood(id, name);
        listFoods();
    }
}
```

eclipse-workspace - MySQLWeek5/src/foods/Appfoods.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Task List
MySQLWeek5 IRE System Library [JavaSE-1.8]
src
  foods
    Appfoods.java
    foodsdao
      DBConnection.java
      foodDao.java
    foods.types
      Food.java
      foods.mysql
  Referenced Libraries
  practice
  practices
  school
  week4
  week4Coding
  week5
  week6
  weeksqlicon
Appfoods.java
72     System.out.print("Enter the Food Id that you want to modify: ");
73     int id = (int) Long.parseLong(scanner.nextLine());
74
75     System.out.print("Enter the new Food name");
76     String name = scanner.nextLine();
77
78     FoodDao.alterFood(id, name);
79     listFoods();
80   }
81
82   private void listFoods() throws SQLException {
83     List<Food> foods = FoodDao.getFoods();
84
85     Foods.stream().forEach(System.out::println);
86
87   }
88
89   private void createFood() throws SQLException {
90     System.out.println("Enter Food name: ");
91     String name = scanner.nextLine();
92     FoodDao.createFood(name);
93
94   }
95
96   private void printInstruction() {
97     System.out.println("Here is the option");
98     System.out.println("1: create a Food");
99     System.out.println("2: Modify Food");
100    System.out.println("3: Food list");
101    System.out.println("4: Delete Food");
102
103  }
104
105
106
107
108
109
110
111
112
Problems Javadoc Declaration Console Coverage
Appfoods [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Nov 1, 2020 2:36:01 PM)
Connection successful!!!!!!!
Outline
  foods
    Appfoods
      scanner : Scanner
      FoodDao : foodDao
      main(String[]) : void
      run() : void
      deleteFood() : void
      alterFood() : void
      listFoods() : void
      createFood() : void
      printInstruction() : void
2:43 PM 11/1/2020
```

eclipse-workspace - MySQLWeek5/src/foods/dao/DBConnection.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Task List
MySQLWeek5 IRE System Library [JavaSE-1.8]
src
  foods
    Appfoods.java
    DBConnection.java
    Food.java
    foodDao.java
    foods.mysql
  Referenced Libraries
  practice
  practices
  school
  week4
  week4Coding
  week5
  week6
  weeksqlicon
DBConnection.java
1 package foods.dao;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 public class DBConnection {
8   private static final String connectionStr = "jdbc:mysql://localhost:3306/foods";
9   private static final String username = "root";
10  private static final String password = "1234";
11  private static DBConnection instance = new DBConnection();
12
13  private Connection conn;
14
15  private DBConnection() {
16  }
17
18  public static DBConnection getInstance() {
19    return instance;
20  }
21
22  public Connection getConnection() throws SQLException {
23    instance.conn = DriverManager.getConnection(connectionStr, username, password);
24    System.out.println("Connection successful!!!!!!!");
25
26    return conn;
27  }
28
29  }
30
31
32
Problems Javadoc Declaration Console Coverage
Appfoods [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Nov 1, 2020 2:36:01 PM)
Connection successful!!!!!!!
Outline
  foods
    DBConnection
      connectionStr : String
      username : String
      password : String
      instance : DBConnection
        conn : Connection
      getInstance() : DBConnection
      getConnection() : Connection
2:43 PM 11/1/2020
```

eclipse-workspace - MySQLWeek5/src/foods/types/Food.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Appfoods.java DBConnection.java Food.java foodDao.java foods.mysql
Package Explorer X
Final project MyFirstProjects MyProject MyProject1 mysql MySQLWeek5 IRE System Library [JavaSE-1.8]
src
  foods
    Appfoods.java
    foodsdao
      DBConnection.java
      foodDao.java
    foods.types
      Food.java
      foods.mysql
  Referenced Libraries
  practice
  practices
  school
  week4
  week4Coding
  week5
  week6
  weeksqlcon
1 package foods.types;
2
3 public class Food {
4     private int foodId;
5     private String name;
6
7     public String getName() {
8         return name;
9     }
10    public void setName(String name) {
11        this.name = name;
12    }
13    public int getFoodId() {
14        return foodId;
15    }
16    public void setFoodId(int foodId) {
17        this.foodId = foodId;
18    }
19
20    @Override
21    public String toString() {
22        return "foodId[" + foodId + ", : " + name + "]";
23    }
24}
25
26
27
28
29
30
```

Problems Javadoc Declaration Console Coverage
Appfoods [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Nov 1, 2020 2:36:01 PM)
Connection successful!!!!!!

Writable Smart Insert 26:43 : 406 2:43 PM 11/1/2020

eclipse-workspace - MySQLWeek5/src/foods/dao/foodDao.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Appfoods.java DBConnection.java Food.java foodDao.java foods.mysql
Package Explorer X
Final project MyFirstProjects MyProject MyProject1 mysql MySQLWeek5 IRE System Library [JavaSE-1.8]
src
  foods
    Appfoods.java
    foodsdao
      DBConnection.java
      foodDao.java
    foods.types
      Food.java
      foods.mysql
  Referenced Libraries
  practice
  practices
  school
  week4
  week4Coding
  week5
  week6
  weeksqlcon
1 package foods.dao;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.util.ArrayList;
8 import java.util.List;
9
10 import foods.types.Food;
11
12 public class foodDao {
13     public void createFood(String name) throws SQLException {
14         String sql = "INSERT INTO Foods(name) VALUE(?)";
15         Connection conn = null;
16         PreparedStatement statement = null;
17
18         try {
19             Conn = DBConnection.getInstance().getConnection();
20             statement = conn.prepareStatement(sql);
21             statement.setString(1, name);
22
23             statement.executeUpdate();
24         } finally{
25             if(statement != null) {
26                 statement.close();
27             }
28             if(conn != null) {
29                 conn.close();
30             }
31         }
32     }
33 }
34
35 public List<Food> getFoods() throws SQLException {
36     String sql = "SELECT * FROM Foods";
37     Connection conn = null;
38     PreparedStatement statement = null;
39     ResultSet rs = null;
40     List<Food> foods = new ArrayList<>();
41 }
```

Problems Javadoc Declaration Console Coverage
Appfoods [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Nov 1, 2020 2:36:01 PM)
Connection successful!!!!!!

Writable Overwrite 121 : 6 : 2596 2:43 PM 11/1/2020

eclipse-workspace - MySQLWeek5/src/foods/dao/foodDao.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Appfoods.java DBConnection.java Food.java foodDao.java foods.mysql
Package Explorer X
Final project
MyFirstProjects
MyProject
MyProject1
mysql
MySQLWeek5
> IRE System Library [JavaSE-1.8]
src
  foods
    Appfoods.java
    foods.dao
      DBConnection.java
      foodDao.java
    foods.types
      Food.java
      foods.mysql
  Referenced Libraries
    practice
    practices
    school
    week4
    week4Coding
    week5
    week6
    weeksqlcon
42  try {
43      conn = DBConnection.getInstance().getConnection();
44      statement = conn.prepareStatement(sql);
45      rs = statement.executeQuery();
46
47      while (rs.next()) {
48          int foodId = rs.getInt("food_id");
49          String name = rs.getString("name");
50          Food seafood = new Food();
51          foods.add(seafood);
52      }
53  } catch (SQLException e) {
54      e.printStackTrace();
55  } finally {
56
57      if(rs != null) {
58          rs.close();
59      }
60      if(statement != null) {
61          statement.close();
62      }
63      if(conn != null) {
64          conn.close();
65      }
66  }
67
68  return foods;
69 }
70
71
72
73
74 public void alterFood(int id, String name) throws SQLException {
75     String sql = "UPDATE foods SET name = ? WHERE food_id = ?";
76     Connection conn = null;
77     PreparedStatement statement = null;
78
79     try {
80         conn = DBConnection.getInstance().getConnection();
81         statement = conn.prepareStatement(sql);
82         statement.setString(1, name);
83         statement.setInt(2, id);
84
85         statement.executeUpdate();
86     } catch (SQLException e) {
87         e.printStackTrace();
88     } finally {
89
90         if(statement != null) {
91             statement.close();
92         }
93         if(conn != null) {
94             conn.close();
95         }
96     }
97 }
98
99
100 public void deleteFood(int foodId) throws SQLException {
101     String sql = "DELETE FROM foods WHERE food_id = ?";
102     Connection conn = null;
103     PreparedStatement statement = null;
104
105     try {
106         conn = DBConnection.getInstance().getConnection();
107         statement = conn.prepareStatement(sql);
108         statement.setInt(1, foodId);
109
110     } catch (SQLException e) {
111         e.printStackTrace();
112     } finally {
113
114 }
```

Task List X

Find All Activate...

Outline X

- foods.dao
- foodDao
 - createFood(String) : void
 - getFoods() : List<Food>
 - alterFood(int, String) : void
 - deleteFood(int) : void

Problems Javadoc Declaration Console Coverage

Appfoods [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Nov 1, 2020 2:36:01 PM)

Connection successful!!!!!!

Writable Overwrite | 121 : 6 : 2596

2:43 PM 11/1/2020

eclipse-workspace - MySQLWeek5/src/foods/dao/foodDao.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Appfoods.java DBConnection.java Food.java foodDao.java foods.mysql
Package Explorer X
Final project
MyFirstProjects
MyProject
MyProject1
mysql
MySQLWeek5
> IRE System Library [JavaSE-1.8]
src
  foods
    Appfoods.java
    foods.dao
      DBConnection.java
      foodDao.java
    foods.types
      Food.java
      foods.mysql
  Referenced Libraries
    practice
    practices
    school
    week4
    week4Coding
    week5
    week6
    weeksqlcon
70 public void alterFood(int id, String name) throws SQLException {
71     String sql = "UPDATE foods SET name = ? WHERE food_id = ?";
72     Connection conn = null;
73     PreparedStatement statement = null;
74
75     try {
76         conn = DBConnection.getInstance().getConnection();
77         statement = conn.prepareStatement(sql);
78         statement.setString(1, name);
79         statement.setInt(2, id);
80
81         statement.executeUpdate();
82     } catch (SQLException e) {
83         e.printStackTrace();
84     } finally {
85
86         if(statement != null) {
87             statement.close();
88         }
89         if(conn != null) {
90             conn.close();
91         }
92     }
93 }
94
95
96
97
98 public void deleteFood(int foodId) throws SQLException {
99     String sql = "DELETE FROM foods WHERE food_id = ?";
100    Connection conn = null;
101    PreparedStatement statement = null;
102
103    try {
104        conn = DBConnection.getInstance().getConnection();
105        statement = conn.prepareStatement(sql);
106        statement.setInt(1, foodId);
107
108    } catch (SQLException e) {
109        e.printStackTrace();
110    } finally {
111
112 }
```

Task List X

Find All Activate...

Outline X

- foods.dao
- foodDao
 - createFood(String) : void
 - getFoods() : List<Food>
 - alterFood(int, String) : void
 - deleteFood(int) : void

Problems Javadoc Declaration Console Coverage

Appfoods [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Nov 1, 2020 2:36:01 PM)

Connection successful!!!!!!

Writable Overwrite | 121 : 6 : 2596

2:43 PM 11/1/2020

eclipse-workspace - MySQLWeek5/src/foods/dao/foodDao.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Task List
MySQLWeek5
src
  foods
    Appfoods.java DBConnection.java Food.java foodDao.java foods.mysql
  final project
  MyFirstProjects
  MyProject
  MyProject1
  mysql
  IRE System Library [JavaSE-1.8]
  week5
    foods
      Appfoods.java
      foodsdao
        DBConnection.java
        foodDao.java
      foods.types
        Food.java
      foods.mysql
    Referenced Libraries
    practice
    practices
    school
    week4
    week4Coding
    week5
    week6
    weeksqlcon
  93 }
  94     if(conn != null) {
  95         conn.close();
  96     }
  97 }
  98 }
  99 }
100 public void deleteFood(int foodId) throws SQLException {
101     String sql = "DELETE FROM foods WHERE food_id = ?";
102     Connection conn = null;
103     PreparedStatement statement = null;
104     try {
105         conn = DBConnection.getInstance().getConnection();
106         statement = conn.prepareStatement(sql);
107         statement.setInt(1, foodId);
108     } catch (SQLException e) {
109         e.printStackTrace();
110     } finally {
111         if(statement != null) {
112             statement.close();
113         }
114         if(conn != null) {
115             conn.close();
116         }
117     }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
459 }
```

Problems Javadoc Declaration Console Coverage

Appfoods [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Nov 1, 2020 2:36:01 PM)

Connection successful!!!!!!

Writable Overwrite 124:1: 2604 2:43 PM 11/1/2020

eclipse-workspace - MySQLWeek5/src/foods/dao/foodDao.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Task List
MySQLWeek5
src
  foods
    Appfoods.java DBConnection.java Food.java foodDao.java foods.mysql
    final project
    MyFirstProjects
    MyProject
    MyProject1
    mysql
    IRE System Library [JavaSE-1.8]
    week5
      foods
        Appfoods.java
        foodsdao
          DBConnection.java
          foodDao.java
        foods.types
          Food.java
        foods.mysql
      Referenced Libraries
      practice
      practices
      school
      week4
      week4Coding
      week5
      week6
      weeksqlcon
      93 }
      94 }
      95 }
      96 }
      97 }
      98 }
      99 }
      100 }
      101 }
      102 }
      103 }
      104 }
      105 }
      106 }
      107 }
      108 }
      109 }
      110 }
      111 }
      112 }
      113 }
      114 }
      115 }
      116 }
      117 }
      118 }
      119 }
      120 }
      121 }
      122 }
      123 }
      124 }
      125 }
      126 }
      127 }
      128 }
      129 }
      130 }
      131 }
      132 }
      133 }
      134 }
      135 }
      136 }
      137 }
      138 }
      139 }
      140 }
      141 }
      142 }
      143 }
      144 }
      145 }
      146 }
      147 }
      148 }
      149 }
      150 }
      151 }
      152 }
      153 }
      154 }
      155 }
      156 }
      157 }
      158 }
      159 }
      160 }
      161 }
      162 }
      163 }
      164 }
      165 }
      166 }
      167 }
      168 }
      169 }
      170 }
      171 }
      172 }
      173 }
      174 }
      175 }
      176 }
      177 }
      178 }
      179 }
      180 }
      181 }
      182 }
      183 }
      184 }
      185 }
      186 }
      187 }
      188 }
      189 }
      190 }
      191 }
      192 }
      193 }
      194 }
      195 }
      196 }
      197 }
      198 }
      199 }
      200 }
      201 }
      202 }
      203 }
      204 }
      205 }
      206 }
      207 }
      208 }
      209 }
      210 }
      211 }
      212 }
      213 }
      214 }
      215 }
      216 }
      217 }
      218 }
      219 }
      220 }
      221 }
      222 }
      223 }
      224 }
      225 }
      226 }
      227 }
      228 }
      229 }
      230 }
      231 }
      232 }
      233 }
      234 }
      235 }
      236 }
      237 }
      238 }
      239 }
      240 }
      241 }
      242 }
      243 }
      244 }
      245 }
      246 }
      247 }
      248 }
      249 }
      250 }
      251 }
      252 }
      253 }
      254 }
      255 }
      256 }
      257 }
      258 }
      259 }
      260 }
      261 }
      262 }
      263 }
      264 }
      265 }
      266 }
      267 }
      268 }
      269 }
      270 }
      271 }
      272 }
      273 }
      274 }
      275 }
      276 }
      277 }
      278 }
      279 }
      280 }
      281 }
      282 }
      283 }
      284 }
      285 }
      286 }
      287 }
      288 }
      289 }
      290 }
      291 }
      292 }
      293 }
      294 }
      295 }
      296 }
      297 }
      298 }
      299 }
      300 }
      301 }
      302 }
      303 }
      304 }
      305 }
      306 }
      307 }
      308 }
      309 }
      310 }
      311 }
      312 }
      313 }
      314 }
      315 }
      316 }
      317 }
      318 }
      319 }
      320 }
      321 }
      322 }
      323 }
      324 }
      325 }
      326 }
      327 }
      328 }
      329 }
      330 }
      331 }
      332 }
      333 }
      334 }
      335 }
      336 }
      337 }
      338 }
      339 }
      340 }
      341 }
      342 }
      343 }
      344 }
      345 }
      346 }
      347 }
      348 }
      349 }
      350 }
      351 }
      352 }
      353 }
      354 }
      355 }
      356 }
      357 }
      358 }
      359 }
      360 }
      361 }
      362 }
      363 }
      364 }
      365 }
      366 }
      367 }
      368 }
      369 }
      370 }
      371 }
      372 }
      373 }
      374 }
      375 }
      376 }
      377 }
      378 }
      379 }
      380 }
      381 }
      382 }
      383 }
      384 }
      385 }
      386 }
      387 }
      388 }
      389 }
      390 }
      391 }
      392 }
      393 }
      394 }
      395 }
      396 }
      397 }
      398 }
      399 }
      400 }
      401 }
      402 }
      403 }
      404 }
      405 }
      406 }
      407 }
      408 }
      409 }
      410 }
      411 }
      412 }
      413 }
      414 }
      415 }
      416 }
      417 }
      418 }
      419 }
      420 }
      421 }
      422 }
      423 }
      424 }
      425 }
      426 }
      427 }
      428 }
      429 }
      430 }
      431 }
      432 }
      433 }
      434 }
      435 }
      436 }
      437 }
      438 }
      439 }
      440 }
      441 }
      442 }
      443 }
      444 }
      445 }
      446 }
      447 }
      448 }
      449 }
      450 }
      451 }
      452 }
      453 }
      454 }
      455 }
      456 }
      457 }
      458 }
      459 }
      459 }
```

Problems Javadoc Declaration Console Coverage

Appfoods [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Nov 1, 2020 2:36:01 PM)

Connection successful!!!!!!

Writable Overwrite 124:1: 2604 2:45 PM 11/1/2020

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: ass file foods

Schemas: foods

```

1 • create database if not exists foods;
2 • drop table if exists foods;
3 • create table foods (
4   food_id INT not null auto_increment,
5   name varchar (50),
6   primary key (food_id)
7 );
8 • insert into foods (food_id, name) values (1, "Sea Food");
9 • insert into foods (food_id, name) values (2, "Vegetables");
10 • insert into foods (food_id, name) values (3, "Grain Beans");

```

Administration Schemas

Information: Schema: foods

Action Output

Time	Action	Message	Duration / Fetch
42 14:32:39	create table foods (food_id INT not null auto_increment, name varchar (50), primary key food_id)	0 row(s) affected	0.032 sec
43 14:32:39	insert into foods (food_id, name) values (1, "Sea Food")	1 row(s) affected	0.000 sec
44 14:32:39	insert into foods (food_id, name) values (2, "Vegetables")	1 row(s) affected	0.016 sec
45 14:32:39	insert into foods (food_id, name) values (3, "Grain Beans")	1 row(s) affected	0.000 sec
46 14:32:48	select * from foods LIMIT 0, 10000	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

245 PM 11/1/2020

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: ass file foods

Schemas: foods

```

1 • select * from foods;
2

```

Administration Schemas

Information: Schema: foods

Result Grid

food_id	name
1	Sea Food
2	Vegetables
3	Grain Beans
NULL	NULL

foods 3 X

Action Output

Time	Action	Message	Duration / Fetch
42 14:32:39	create table foods (food_id INT not null auto_increment, name varchar (50), primary key food_id)	0 row(s) affected	0.032 sec
43 14:32:39	insert into foods (food_id, name) values (1, "Sea Food")	1 row(s) affected	0.000 sec
44 14:32:39	insert into foods (food_id, name) values (2, "Vegetables")	1 row(s) affected	0.016 sec
45 14:32:39	insert into foods (food_id, name) values (3, "Grain Beans")	1 row(s) affected	0.000 sec
46 14:32:48	select * from foods LIMIT 0, 10000	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

245 PM 11/1/2020